AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph at page 4, lines 6-23, with the following amended paragraph:

--Briefly, the present invention provides a method, system and infrastructure that allow an application to run with specified versions of components bound thereto, wherein each component (also referred to herein as an assembly, wherein an assembly is a set of one or more component files which are versioned and [[ship]] shipped as a unit) may exist and run side-by-side on the system with other versions of the same assembly being used by other applications. To this end, the application provides a manifest to specify any desired assembly versions. An activation context is created for the application based on the manifest to map global, version independent names to a particular version maintained in an assembly cache or in the application's respective directory. In order to specify a version for an assembly, applications need not have their code rewritten, but instead can provide the application manifest, (e.g., in an XML data structure). Similarly, each assembly has an assembly manifest that specifies the versions of dependent assemblies.--

Please replace the paragraph at page 5, lines 1-10, with the following amended paragraph:

--manifest in the same file system directory as the calling executable. When an application manifest exists, the operating system checks for an activation context for the application that was built from the manifest. If the activation context does not exist (for example this is the first time application has been executed), or it exists but is not coherent with current policy, an activation context is created via the application manifest and the assembly manifests of any dependent assemblies/assemblies listed in the application manifest.--

Please replace the paragraph at page 10, lines 23-25, with the following amended paragraph:

--The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. 1 illustrates a hard disk drive [[140]] 141 that--

Please replace the paragraph at page 12, lines 1-18, with the following amended paragraph:

--145, other program modules 146, and program data 147 are given different numbers herein to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer [[20]] 110 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through [[a]] an output peripheral interface [[190]] 195.--

Please replace the paragraph at page 31, lines 20-25, with the following amended paragraph:

--When the application API 300_1 [[receive]] receives the request, the request data (e.g., the application provided name) is passed to a runtime version-matching mechanism 502 (the arrow labeled two (2)). The runtime version-matching mechanism 502 locates the correct activation context 304_1 for the calling application 202_1, and accesses the records therein to determine the correct--

Please replace the paragraph at page 1, lines 5-10, with the following amended paragraph:

The present application claims priority to U.S. Provisional Patent Application Ser. No. 60/199,374, filed Apr. 24, 2000, and is also related to copending U.S. patent application entitled "Configurations for Binding Software Assemblies to Applications," Ser. No. 09/842,278, filed concurrently herewith.

Please replace the paragraph at page 16, lines 5-25, with the following amended paragraph:

Although the application manifest lists its dependencies on assembly versions, it should be noted that technically, the application is only dependent on the manifest-specified assemblies themselves, not necessarily the exact versions that are specified. Instead, the specified versions are only those which have been

tested with the application. In one implementation, another version of a needed assembly can replace the application-preferred version when additional information is available to the assembly loading mechanism (e.g., an operating system component). For example, as also described in copending U.S. patent application entitled "Configurations for Binding Software Assemblies to Applications" Ser. No. 09/842,278, assigned to the assignee of the present application, filed concurrently herewith, and hereby incorporated by reference, policy data (configurations) and/or an assembly manifest associated with the specified assembly may override the version information set forth in the application manifest. Nevertheless, it is expected that such policies and assembly manifests will be used conservatively, e.g., only for important fixes and/or after thorough testing, since they will cause replacement of the trusted assembly version.

Please replace the paragraph at page 25, line 18-page 26, line 15, with the following amended paragraph:

Note that although not shown in FIG. 2, a later-installed application policy can also change an application manifest's requested assembly version such that the existing manifest does not have to be replaced or have its contents modified to allow an application to effectively change what is in its original (e.g., shipped and installed with the application code) manifest. In other words, via the application policy, the original manifest does not have to be reinstalled or modified to change a dependency. The application policy settings can also bypass an assembly manifest's version overrides, and a system policy can override any other versioning metadata. In this manner, a flexible architecture is provided in which application authors and assembly authors, as well as system administrators, may create safe, isolated applications by simply creating declarative manifests or policies that control the dependencies on shared assemblies, while enabling changes to dependencies to be made when necessary. These and additional aspects of the architecture, including methods for resolving which version will be loaded when the various manifests and policies provide different instructions, are further described in the aforementioned U.S. patent application Ser. No. 09/842,278, entitled "Configurations for Binding Software Assemblies to Applications."